Project MoIB

MoIB = Mpi_vm over IB

- A cluster of virtual machines for parallel applications in MPI

Hyunwoo KIM, October 5 2011

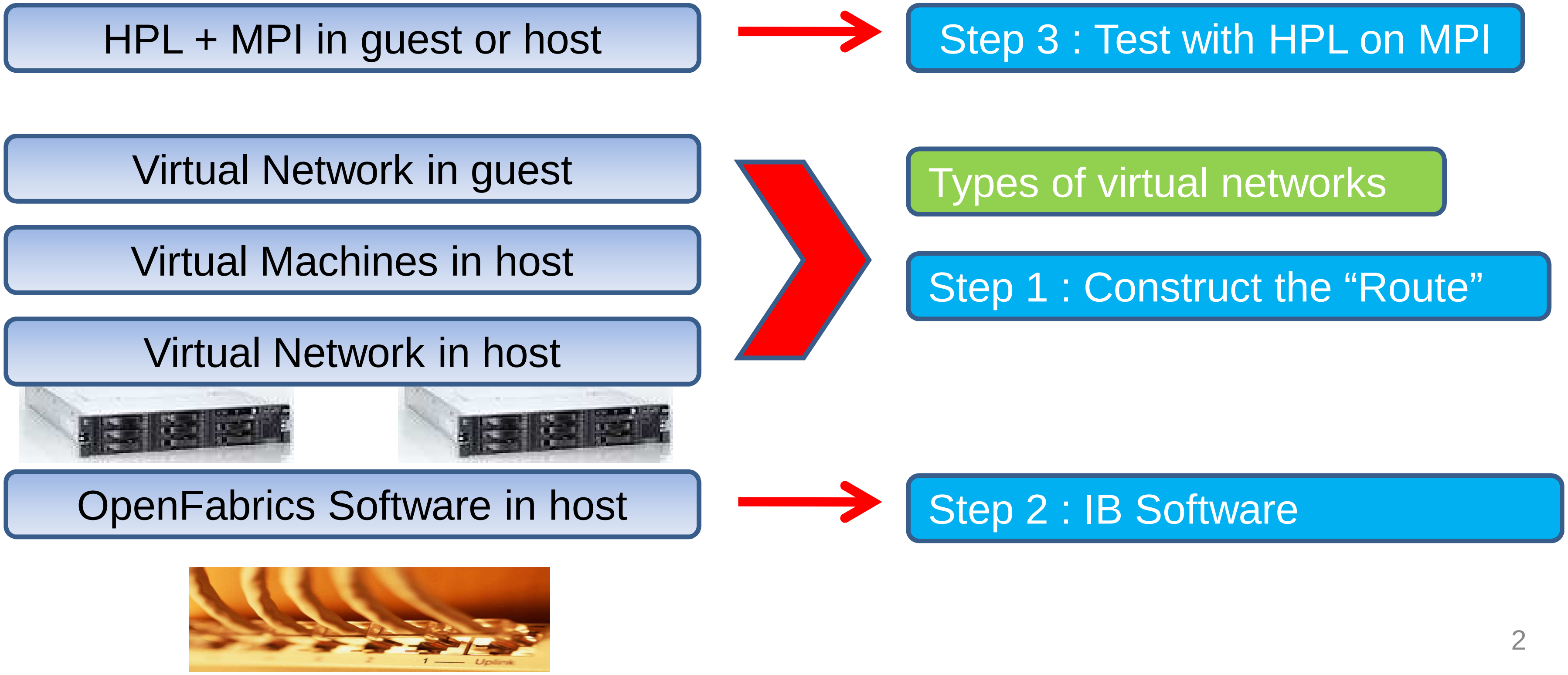
- Supervised by Steve TIMM
- Advised by Dan YOCUM, Faarooq LOWE

Motivation and Configuration

A cluster of “virtual machines” for parallel applications(LQCD group)

Building this type of virtual cluster requires

- Install and configure usual parallel environments
  - IB software(OFED) + MPI library + Application(HPL)
- Building a route between MPI processes on VMs
  - virtual machines + virtual network



HPL + MPI in guest or host

Step 3 : Test with HPL on MPI

Virtual Network in guest

Virtual Machines in host

Virtual Network in host

Types of virtual networks

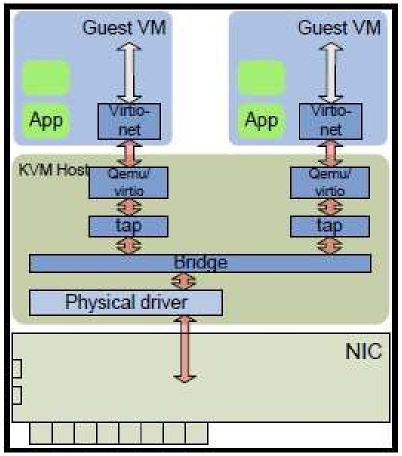
Step 1 : Construct the “Route”

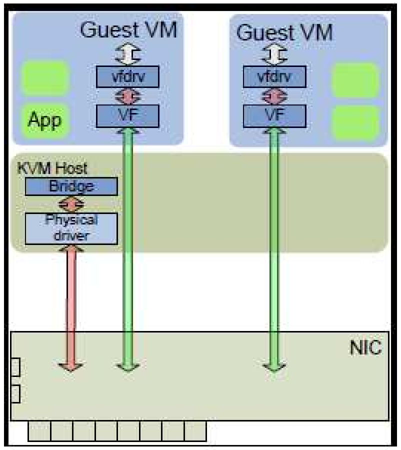
OpenFabrics Software in host

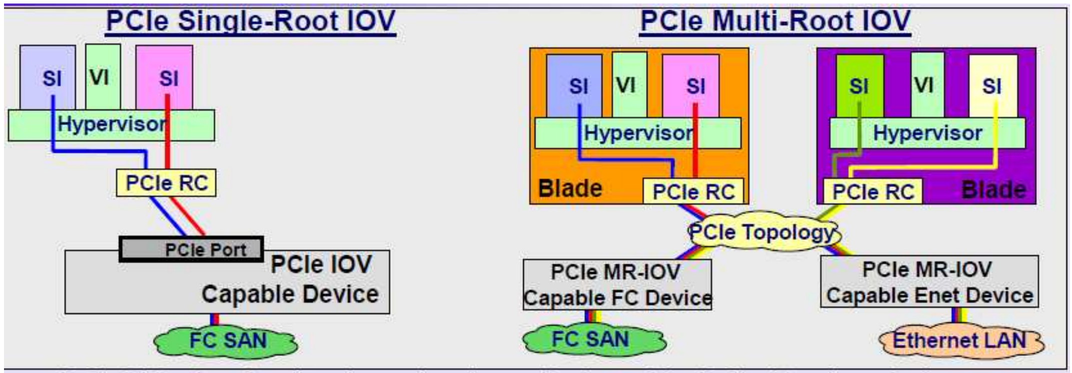
Step 2 : IB Software

Types of virtual networks

Network Virtualization="Sharing of Network Resources"
Software vs Hardware-based Sharing : SRIOV







- PCI-SIG standardized mechanisms that enable PCIe Devices to be directly shared, with no run-time overheads:
  - ✓ Single-Root IOV - Direct sharing between SIs on a single system
  - ✓ Multi-Root IOV - Direct sharing between SIs on multiple systems

SRIOV: Make a physical device appear as multiple virtual devices

Virtual network solution better than software-based approach

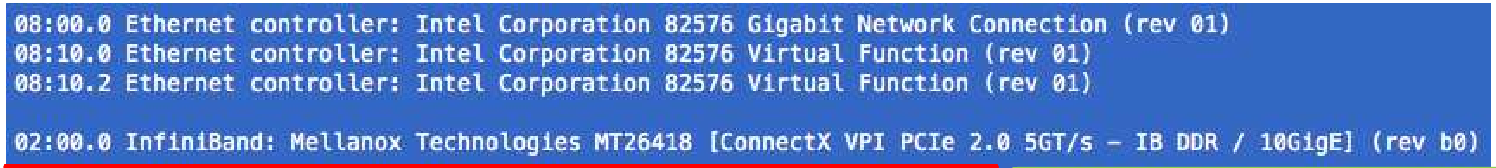
- Less burden in hypervisor, less CPU consumption, more scalability

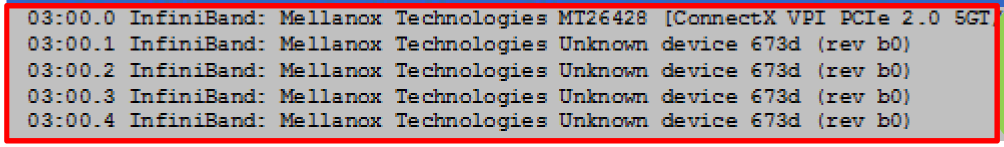
Question: Does Mellanox adapter in fcl017 support SRIOV?

- Mlx brochures : say yes, but emails to Mlx engineers unanswered

Question: Assuming it does, how do we enable SRIOV?

- Intel 82576 GbE Controllers support SRIOV: modprobe igb max_vfs=2

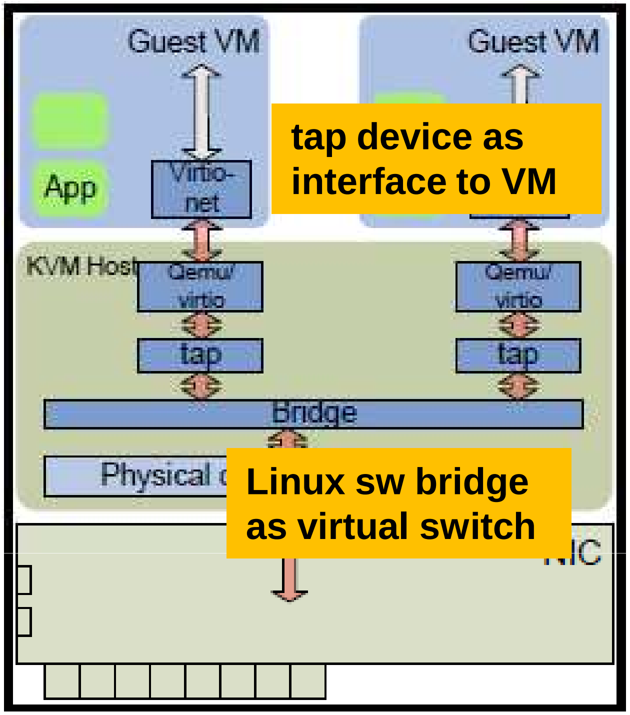

```
08:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
08:10.0 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
08:10.2 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)

02:00.0 InfiniBand: Mellanox Technologies MT26418 [ConnectX VPI PCIe 2.0 5GT/s - IB DDR / 10GigE] (rev b0)
```


```
03:00.0 InfiniBand: Mellanox Technologies MT26428 [ConnectX VPI PCIe 2.0 5GT
03:00.1 InfiniBand: Mellanox Technologies Unknown device 673d (rev b0)
03:00.2 InfiniBand: Mellanox Technologies Unknown device 673d (rev b0)
03:00.3 InfiniBand: Mellanox Technologies Unknown device 673d (rev b0)
03:00.4 InfiniBand: Mellanox Technologies Unknown device 673d (rev b0)
```

Presentation of Mellanox engineer at 2010 OpenFabrics workshp
Asked how in email: no answer

Types of virtual networks

Software-based Sharing: Constructing the “Path”

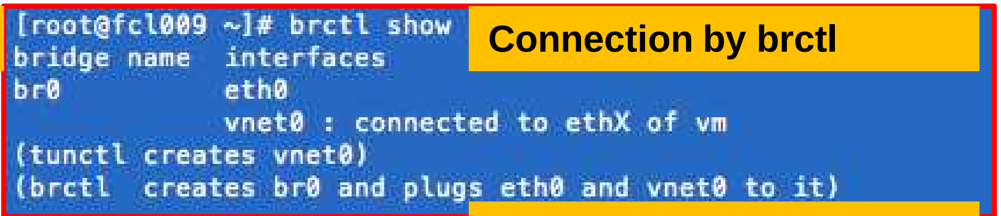


tap device as interface to VM

Linux sw bridge as virtual switch

Common: bridge created by brctl, interfaces by tunctl
Connection btn interface to vm and interface to IB

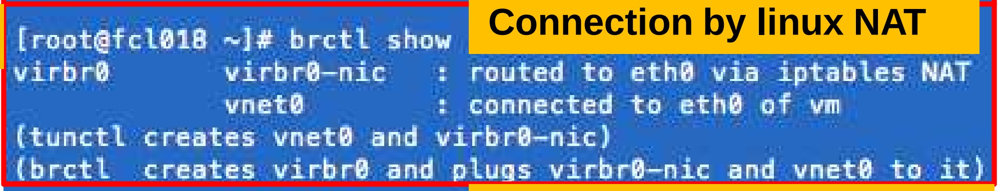
Bridged



Connection by brctl

```
[root@fcl009 ~]# brctl show
bridge name  interfaces
br0          eth0
             vnet0 : connected to ethX of vm
(tunctl creates vnet0)
(brctl  creates br0 and plugs eth0 and vnet0 to it)
```

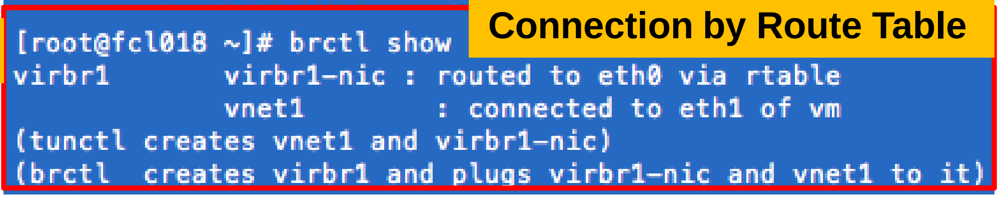
NAT



Connection by linux NAT

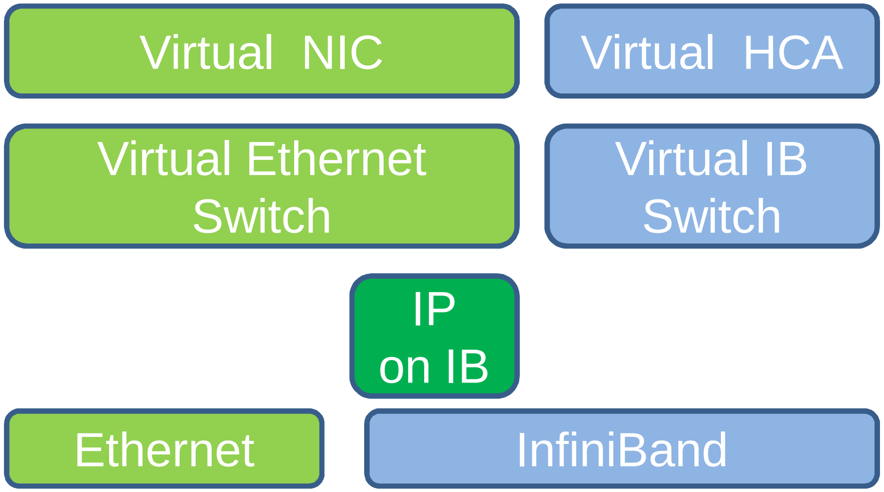
```
[root@fcl018 ~]# brctl show
virbr0       virbr0-nic   : routed to eth0 via iptables NAT
             vnet0        : connected to eth0 of vm
(tunctl creates vnet0 and virbr0-nic)
(brctl  creates virbr0 and plugs virbr0-nic and vnet0 to it)
```

Route



Connection by Route Table

```
[root@fcl018 ~]# brctl show
virbr1       virbr1-nic : routed to eth0 via rtable
             vnet1        : connected to eth1 of vm
(tunctl creates vnet1 and virbr1-nic)
(brctl  creates virbr1 and plugs virbr1-nic and vnet1 to it)
```



Virtual NIC

Virtual HCA

Virtual Ethernet Switch

Virtual IB Switch

IP on IB

Ethernet

InfiniBand

1. Create the virtual network
   - Virtual Switch
2. Create IPoIB
3. Construct the rtable
   - ip route add
4. SSH for Private Network

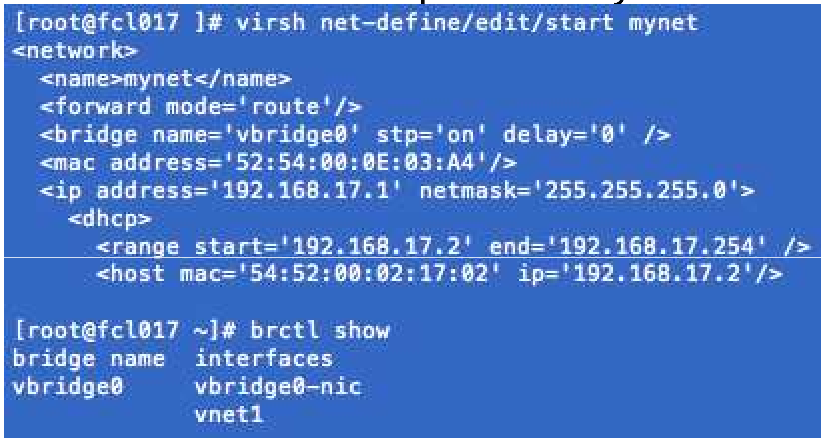
Task 1 : Construct the “Route”

1. Create Route Mode Virtual Network
2. Create IPoIB : ifcfg-ib0

Define Route mode virtual network

Linux bridge created by brctl

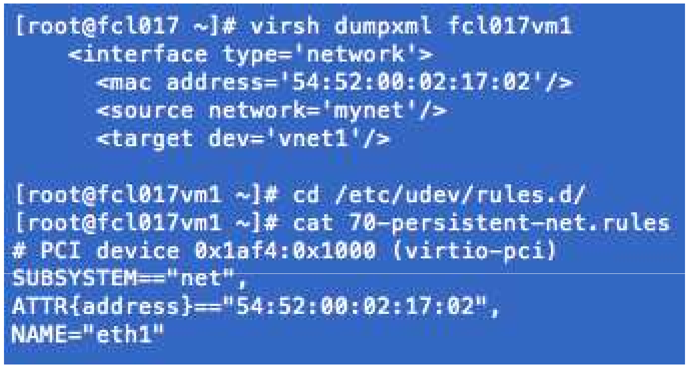
Interface to VM as tap device by tunctl



```
[root@fcl017 ]# virsh net-define/edit/start mynet
<network>
  <name>mynet</name>
  <forward mode='route'/>
  <bridge name='vbridge0' stp='on' delay='0' />
  <mac address='52:54:00:0E:03:A4'/>
  <ip address='192.168.17.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.17.2' end='192.168.17.254' />
      <host mac='54:52:00:02:17:02' ip='192.168.17.2'/>

[root@fcl017 ~]# brctl show
bridge name   interfaces
vbridge0      vbridge0-nic
              vnet1
```

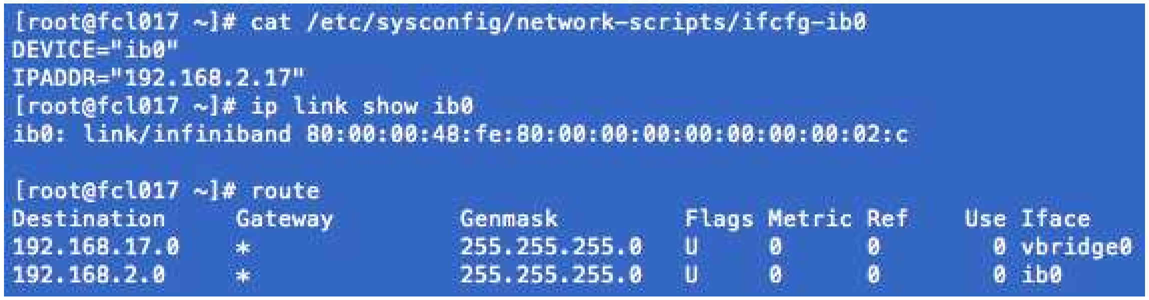
Define VM to connect to bridge

MAC addr in XML is used in VM’s eth1



```
[root@fcl017 ~]# virsh dumpxml fcl017vm1
    <interface type='network'>
      <mac address='54:52:00:02:17:02'/>
      <source network='mynet'/>
      <target dev='vnet1'/>

[root@fcl017vm1 ~]# cd /etc/udev/rules.d/
[root@fcl017vm1 ~]# cat 70-persistent-net.rules
# PCI device 0x1af4:0x1000 (virtio-pci)
SUBSYSTEM=="net",
ATTR{address}=="54:52:00:02:17:02",
NAME="eth1"
```

In hosts we create a new interface(ib0) to IB(ifcfg-ib0 handled by IPoIB driver)
This physical interface ib0 creates a subnet 192.168.2.0 between two hosts
The bridge creates a subnet 17.0 (shown in routing table)

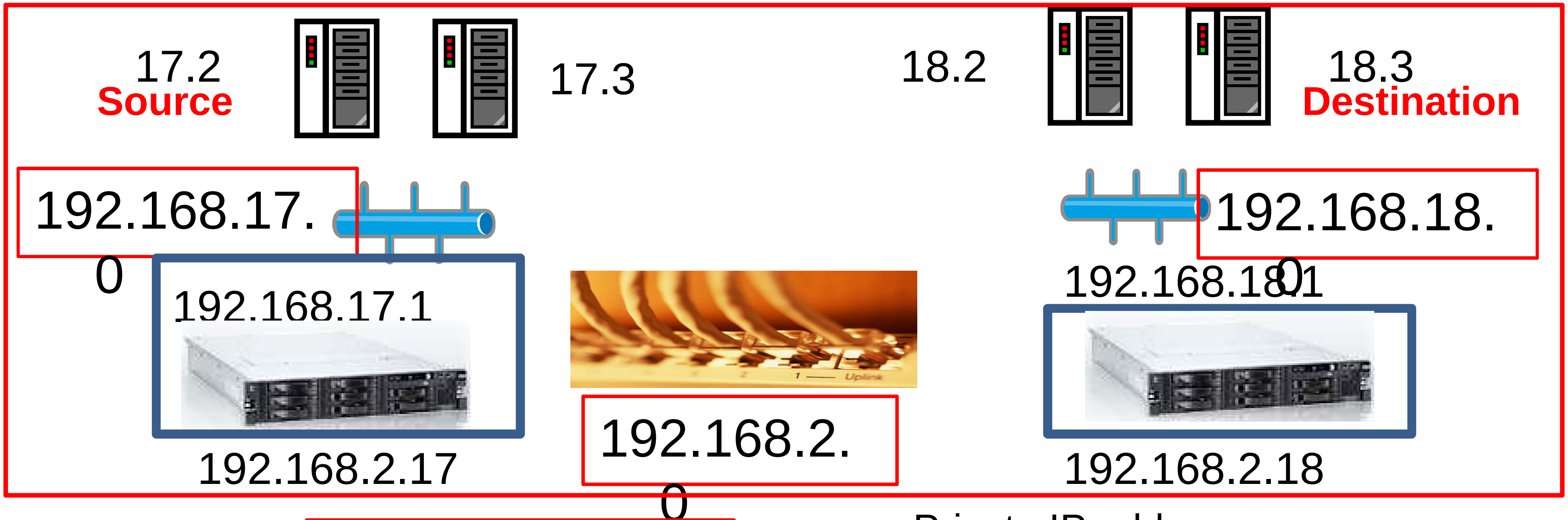


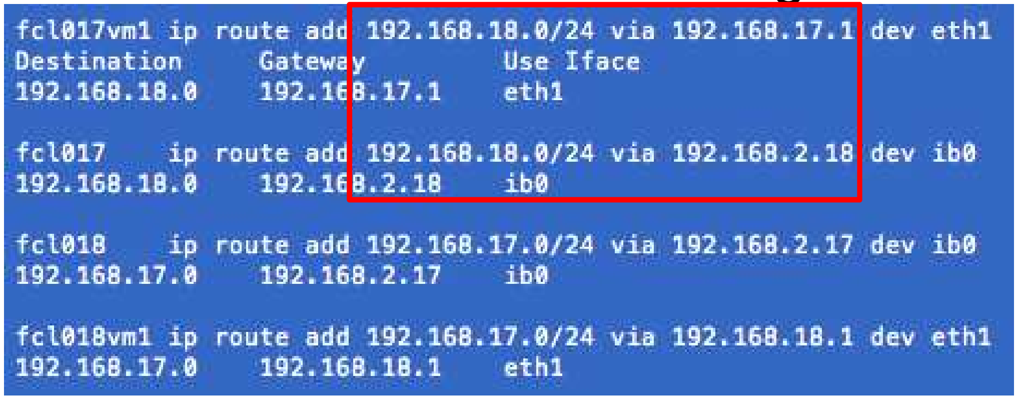
```
[root@fcl017 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ib0
DEVICE="ib0"
IPADDR="192.168.2.17"
[root@fcl017 ~]# ip link show ib0
ib0: link/infiniband 80:00:00:48:fe:80:00:00:00:00:00:00:02:c

[root@fcl017 ~]# route
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.17.0    *               255.255.255.0   U     0      0        0 vbridge0
192.168.2.0     *               255.255.255.0   U     0      0        0 ib0
```

Now we need routing tables

Task 1 : Construct the “Route”

3. Routing Table: Finding the next hop
4. SSH Configuration for PK Authentication

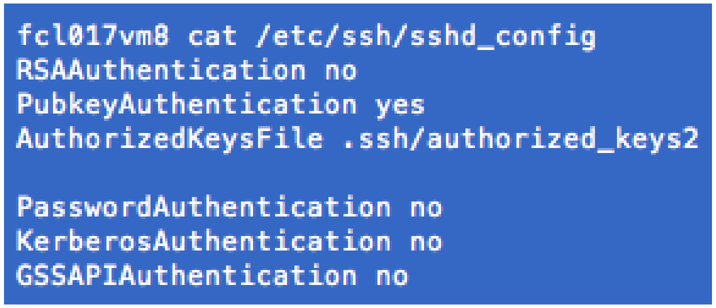




```
fcl017vm1 ip route add 192.168.18.0/24 via 192.168.17.1 dev eth1
Destination     Gateway         Use Iface
192.168.18.0    192.168.17.1    eth1

fcl017    ip route add 192.168.18.0/24 via 192.168.2.18 dev ib0
192.168.18.0    192.168.2.18    ib0

fcl018    ip route add 192.168.17.0/24 via 192.168.2.17 dev ib0
192.168.17.0    192.168.2.17    ib0

fcl018vm1 ip route add 192.168.17.0/24 via 192.168.18.1 dev eth1
192.168.17.0    192.168.18.1    eth1
```

Private IP address

PublicKey Auth in sshd_config



```
fcl017vm8 cat /etc/ssh/sshd_config
RSAAuthentication no
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys2

PasswordAuthentication no
KerberosAuthentication no
GSSAPIAuthentication no
```

Static routes in /etc/sysconfig/network-scripts/route-eth1 (or ib0)

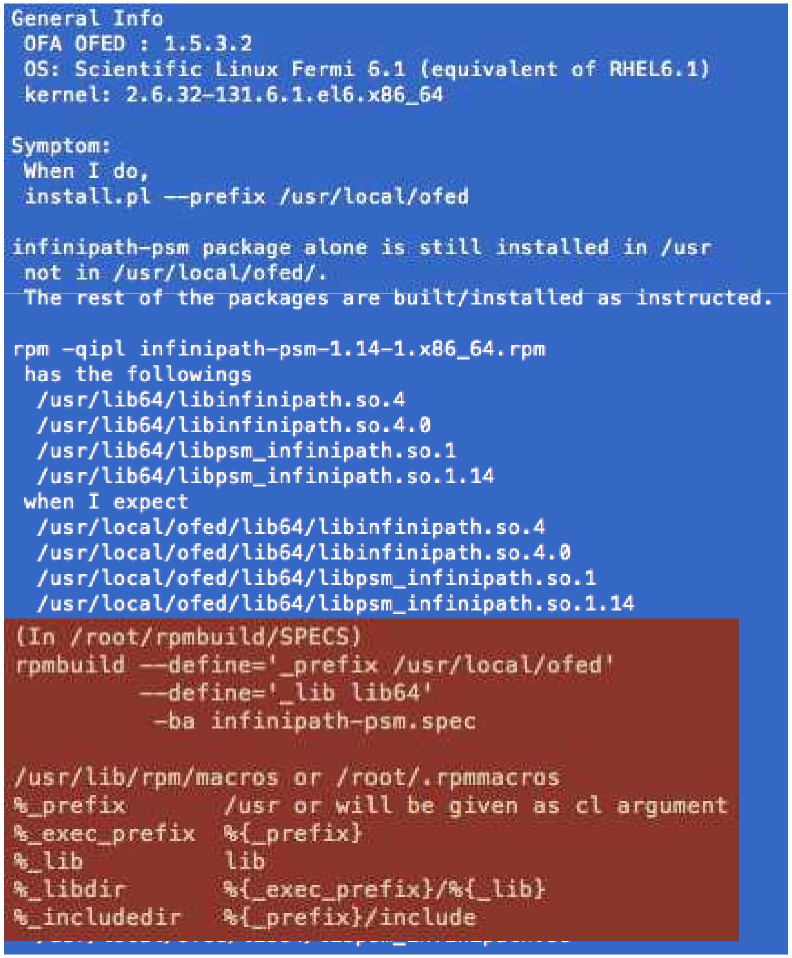
Task 2 : InfiniBand Software

- Software from OpenFabrics Alliance
  - Use Mellanox version for firmware update
- The command: install.pl --prefix /usr/local/ofed
- Intensive hacking of this perl script
  - needed to make sure the install was OK
  - rpm -ivh package.src.rpm
  - package compile at /root/rpmbuild/BUILD
  - rpmbuild -> package.x86_64.rpm
  - rpm -ivh package.x86_64.rpm
- All packages processed OK, except for one infinipath-psm from the package list

Task 2 : IB Software

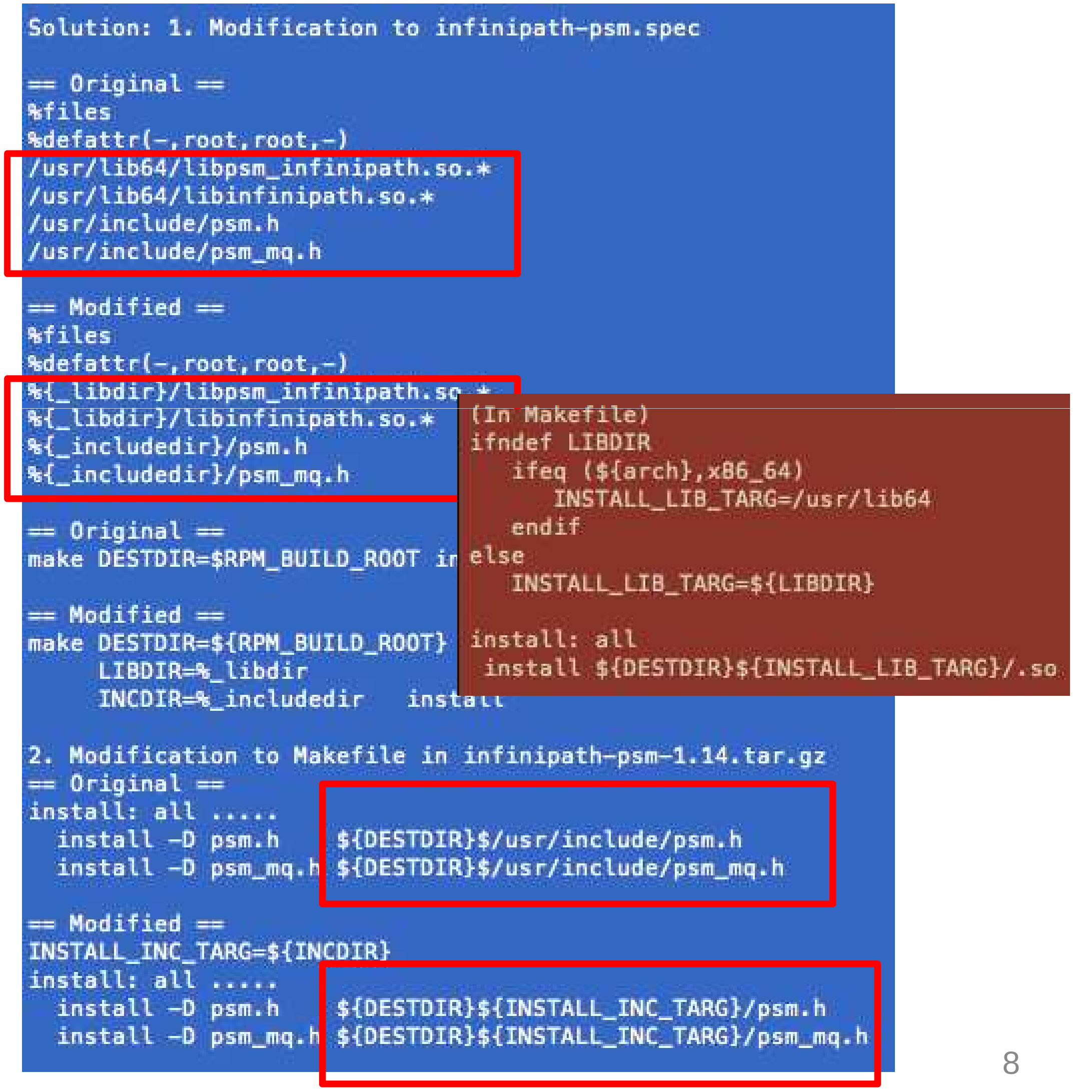
Hacking infinipath-psm

Hardcoded paths for lib and include, replaced by use of rpm macros, Makefile variable and command line options to make



```
General Info
 OFA OFED : 1.5.3.2
 OS: Scientific Linux Fermi 6.1 (equivalent of RHEL6.1)
 kernel: 2.6.32-131.6.1.el6.x86_64

Symptom:
 When I do,
 install.pl --prefix /usr/local/ofed

infinipath-psm package alone is still installed in /usr
 not in /usr/local/ofed/.
 The rest of the packages are built/installed as instructed.

rpm -qipl infinipath-psm-1.14-1.x86_64.rpm
 has the followings
  /usr/lib64/libinfinipath.so.4
  /usr/lib64/libinfinipath.so.4.0
  /usr/lib64/libpsm_infinipath.so.1
  /usr/lib64/libpsm_infinipath.so.1.14
 when I expect
  /usr/local/ofed/lib64/libinfinipath.so.4
  /usr/local/ofed/lib64/libinfinipath.so.4.0
  /usr/local/ofed/lib64/libpsm_infinipath.so.1
  /usr/local/ofed/lib64/libpsm_infinipath.so.1.14
```

```
(In /root/rpmbuild/SPECS)
rpmbuild --define='_prefix /usr/local/ofed'
         --define='_lib lib64'
          -ba infinipath-psm.spec

/usr/lib/rpm/macros or /root/.rpmmacros
%_prefix        /usr or will be given as cl argument
%_exec_prefix   %{_prefix}
%_lib           lib
%_libdir        %{_exec_prefix}/%{_lib}
%_includedir    %{_prefix}/include
```

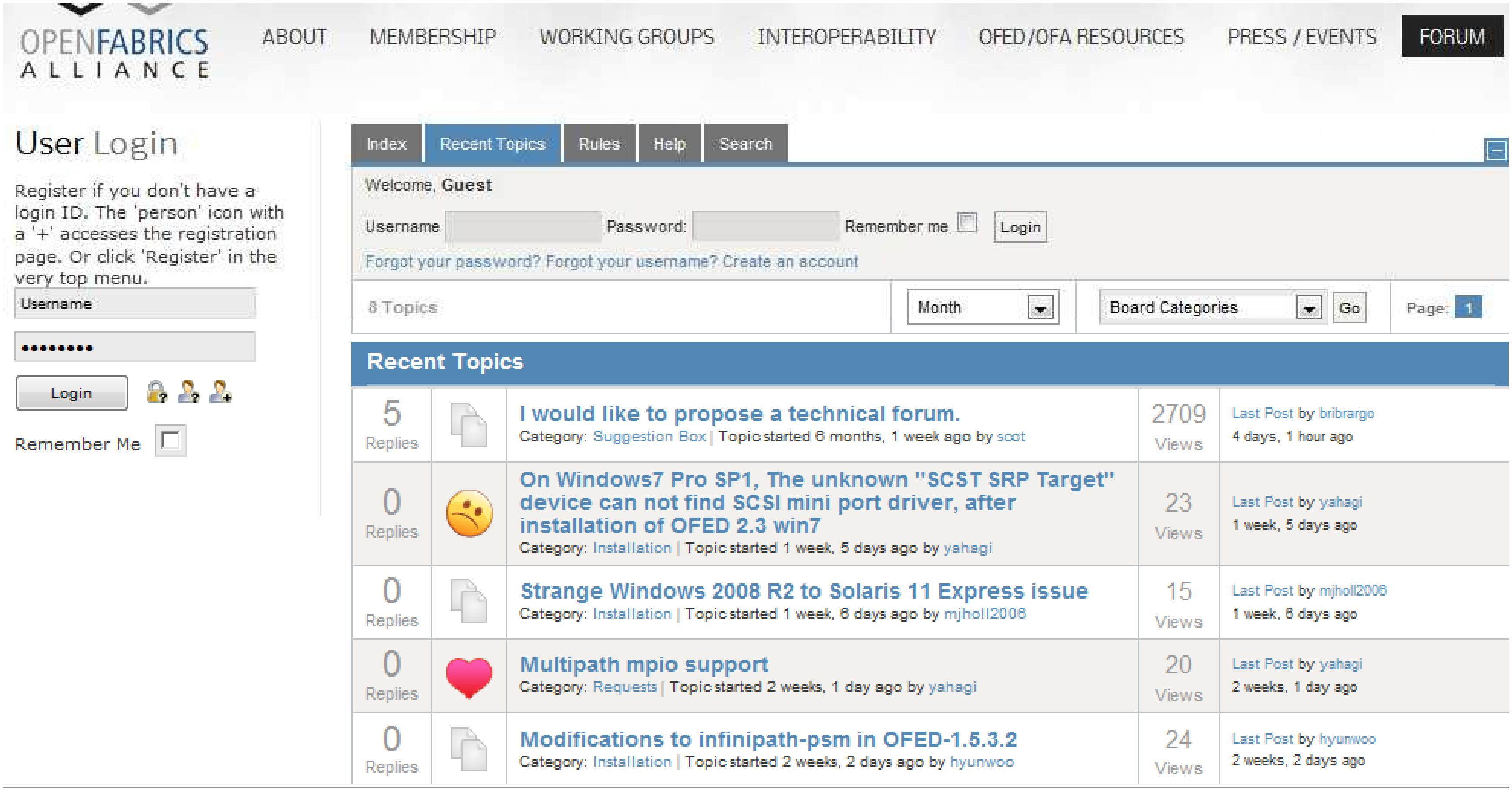


```
Solution: 1. Modification to infinipath-psm.spec

== Original ==
%files
%defattr(-,root,root,-)
/usr/lib64/libpsm_infinipath.so.*
/usr/lib64/libinfinipath.so.*
/usr/include/psm.h
/usr/include/psm_mq.h

== Modified ==
%files
%defattr(-,root,root,-)
%{_libdir}/libpsm_infinipath.so.*
%{_libdir}/libinfinipath.so.*
%{_includedir}/psm.h
%{_includedir}/psm_mq.h

== Original ==
make DESTDIR=$RPM_BUILD_ROOT in

== Modified ==
make DESTDIR=${RPM_BUILD_ROOT}
     LIBDIR=%_libdir
     INCDIR=%_includedir   install

2. Modification to Makefile in infinipath-psm-1.14.tar.gz
== Original ==
install: all .....
  install -D psm.h    ${DESTDIR}$/usr/include/psm.h
  install -D psm_mq.h ${DESTDIR}$/usr/include/psm_mq.h

== Modified ==
INSTALL_INC_TARG=${INCDIR}
install: all .....
  install -D psm.h    ${DESTDIR}${INSTALL_INC_TARG}/psm.h
  install -D psm_mq.h ${DESTDIR}${INSTALL_INC_TARG}/psm_mq.h
```

```
(In Makefile)
ifndef LIBDIR
   ifeq (${arch},x86_64)
      INSTALL_LIB_TARG=/usr/lib64
   endif
else
   INSTALL_LIB_TARG=${LIBDIR}

install: all
 install ${DESTDIR}${INSTALL_LIB_TARG}/.so
```

Task 2 : IB Software

Reported to OFA Forum Modifications to infinipath-psm



User Login

Register if you don't have a login ID. The 'person' icon with a '+' accesses the registration page. Or click 'Register' in the very top menu.

Username

•••••••••

Login

Remember Me □

Welcome, **Guest**

Username Password: Remember me □ Login

Forgot your password? Forgot your username? Create an account

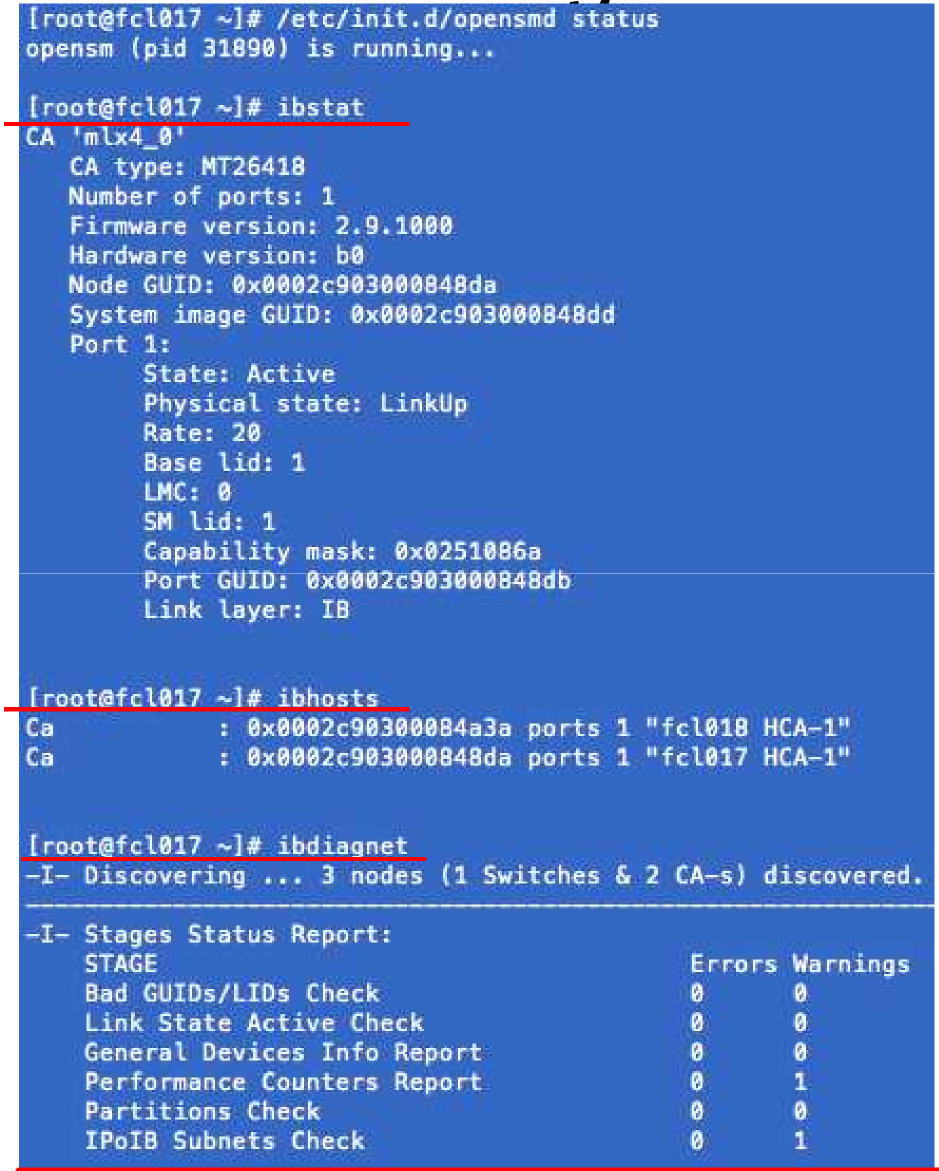
8 Topics | Month | Board Categories | Go | Page: 1

Recent Topics

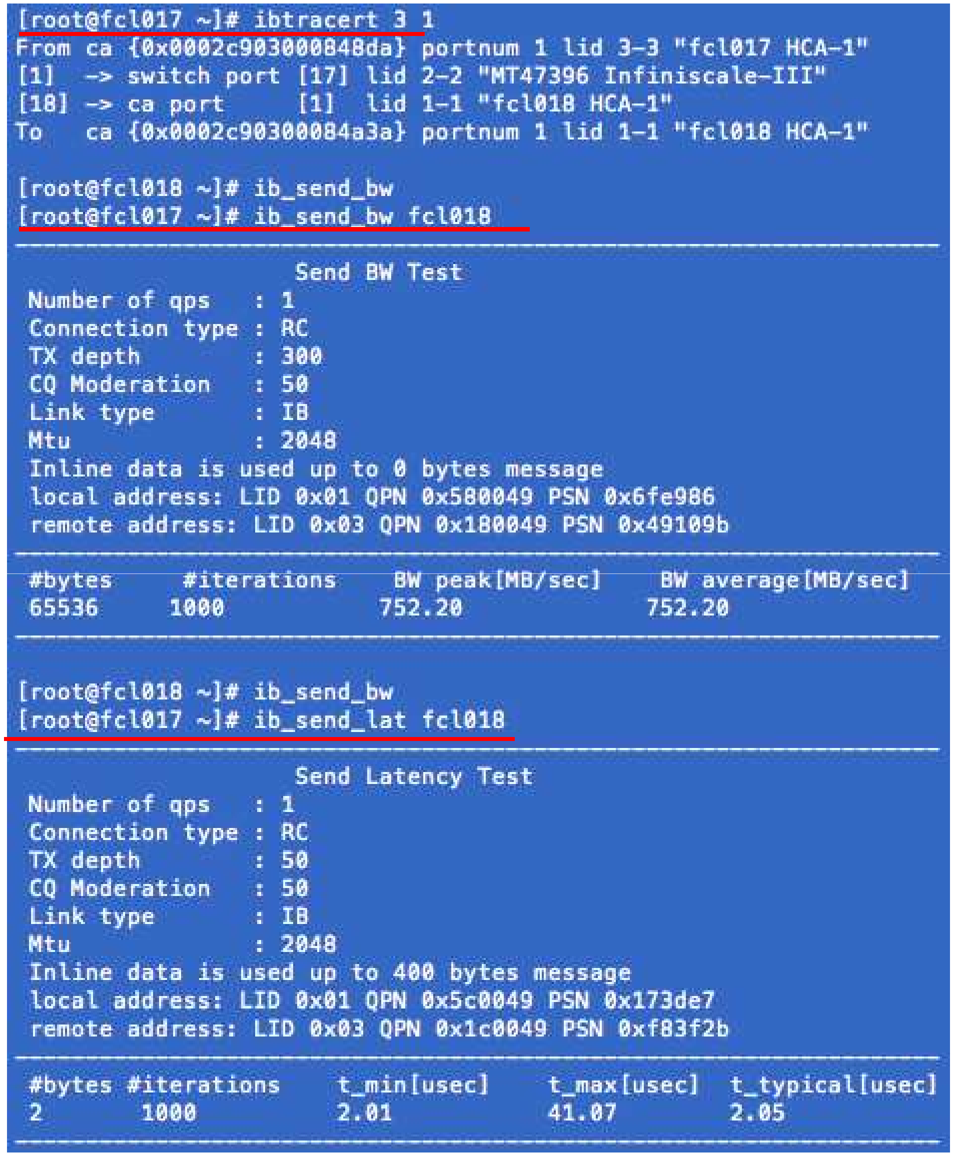
Replies	Topic	Views	Last Post
5 Replies	**I would like to propose a technical forum.** Category: Suggestion Box | Topic started 6 months, 1 week ago by scot	2709 Views	Last Post by bribrargo 4 days, 1 hour ago
0 Replies	**On Windows7 Pro SP1, The unknown "SCST SRP Target" device can not find SCSI mini port driver, after installation of OFED 2.3 win7** Category: Installation | Topic started 1 week, 5 days ago by yahagi	23 Views	Last Post by yahagi 1 week, 5 days ago
0 Replies	**Strange Windows 2008 R2 to Solaris 11 Express issue** Category: Installation | Topic started 1 week, 6 days ago by mjholl2006	15 Views	Last Post by mjholl2006 1 week, 6 days ago
0 Replies	**Multipath mpio support** Category: Requests | Topic started 2 weeks, 1 day ago by yahagi	20 Views	Last Post by yahagi 2 weeks, 1 day ago
0 Replies	**Modifications to infinipath-psm in OFED-1.5.3.2** Category: Installation | Topic started 2 weeks, 2 days ago by hyunwoo	24 Views	Last Post by hyunwoo 2 weeks, 2 days ago

Task 2 : IB Software

IB Diagnostic Tools in the OFA



```
[root@fcl017 ~]# /etc/init.d/opensmd status
opensm (pid 31890) is running...

[root@fcl017 ~]# ibstat
CA 'mlx4_0'
   CA type: MT26418
   Number of ports: 1
   Firmware version: 2.9.1000
   Hardware version: b0
   Node GUID: 0x0002c903000848da
   System image GUID: 0x0002c903000848dd
   Port 1:
        State: Active
        Physical state: LinkUp
        Rate: 20
        Base lid: 1
        LMC: 0
        SM lid: 1
        Capability mask: 0x0251086a
        Port GUID: 0x0002c903000848db
        Link layer: IB


[root@fcl017 ~]# ibhosts
Ca             : 0x0002c90300084a3a ports 1 "fcl018 HCA-1"
Ca             : 0x0002c903000848da ports 1 "fcl017 HCA-1"


[root@fcl017 ~]# ibdiagnet
-I- Discovering ... 3 nodes (1 Switches & 2 CA-s) discovered.

-I- Stages Status Report:
    STAGE                                     Errors Warnings
    Bad GUIDs/LIDs Check                      0      0
    Link State Active Check                   0      0
    General Devices Info Report               0      0
    Performance Counters Report               0      1
    Partitions Check                          0      0
    IPoIB Subnets Check                       0      1
```



```
[root@fcl017 ~]# ibtracert 3 1
From ca {0x0002c903000848da} portnum 1 lid 3-3 "fcl017 HCA-1"
[1]  -> switch port [17] lid 2-2 "MT47396 Infiniscale-III"
[18] -> ca port     [1]  lid 1-1 "fcl018 HCA-1"
To   ca {0x0002c90300084a3a} portnum 1 lid 1-1 "fcl018 HCA-1"

[root@fcl018 ~]# ib_send_bw
[root@fcl017 ~]# ib_send_bw fcl018
------------------------------------------------------------------
                    Send BW Test
 Number of qps   : 1
 Connection type : RC
 TX depth        : 300
 CQ Moderation   : 50
 Link type       : IB
 Mtu             : 2048
 Inline data is used up to 0 bytes message
 local address: LID 0x01 QPN 0x580049 PSN 0x6fe986
 remote address: LID 0x03 QPN 0x180049 PSN 0x49109b
------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]
 65536      1000           752.20             752.20
------------------------------------------------------------------

[root@fcl018 ~]# ib_send_bw
[root@fcl017 ~]# ib_send_lat fcl018
------------------------------------------------------------------
                    Send Latency Test
 Number of qps   : 1
 Connection type : RC
 TX depth        : 50
 CQ Moderation   : 50
 Link type       : IB
 Mtu             : 2048
 Inline data is used up to 400 bytes message
 local address: LID 0x01 QPN 0x5c0049 PSN 0x173de7
 remote address: LID 0x03 QPN 0x1c0049 PSN 0xf83f2b
------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]
 2       1000          2.01           41.07        2.05
------------------------------------------------------------------
```

All results look ok and compared with results of lattice QCD (Amitoj Singh)

Task 3 : Test with HPL on MPI

MPI Library Install and mpirun

- BM use MPI in OFA, but for VM, it’s redundant
  – Use standalone MPI: OpenMPI
- Choose network fabric and interface
  – mpirun --mca btl openib(host), tcp(guest)
  – mpirun --mca btl_tcp_include ib0(host), eth1(guest)
- mpirun : an error trying to initialize IB devices
  – How to increase memlock limit?
  – For root, /etc/security/limits.conf
  – For me, ulimit –l unlimited, where?
    - /etc/init.d/sshd : have to restart sshd sometimes

Task 3 : Test with HPL on MPI

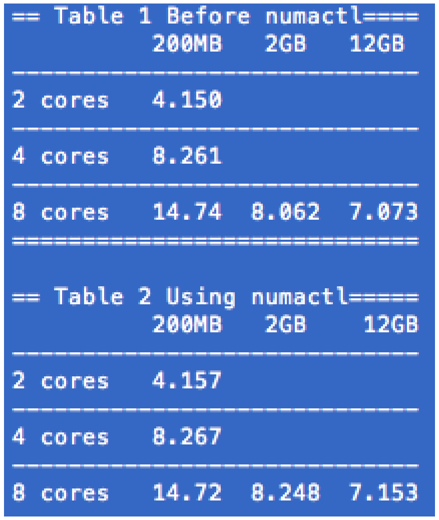
The Cluster is Ready for Tests

- 16 virtual machines on fcl17 and fcl18
  - 8 physical cores on each BM, HyperThread off
  - All VMs has OpenMPI installed
  - Passwordless ssh between each guests
- Hosts have OFA package for InfiniBand
  - How to connect two different fabrics?
    - In Data Link Layer, not possible
    - In Network Layer, IPoIB: IP addr to interface to IB
  - Virtual bridge in “route mode”: VM plugged
  - Host routing tables relay the packets
- Now move on to running MPI applications

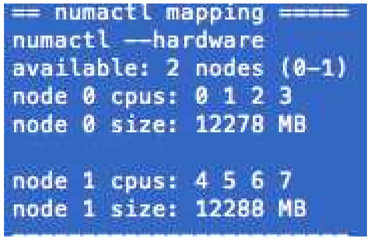
Task 3 : Test with HPL on MPI

Understanding HPL: Test In Bare Metals

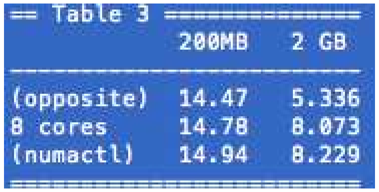
- Measure the MPI performance in FLoatingpoint Operations Per Second
- Question : When do we gain better result?
  - when increasing the number of cores and nodes
- Tests with increasing problem sizes within the total capacity of memory
  - 24 GB (4GB X 6 DIMMs) in fcl0xx machines
  - If smaller problem requires 240 MB, 30 MB will be processed by each of 8 cores
  - If larger problem requires 24 GB, 3 GB will be processed by each of 8 cores
  - Larger problem size produces not good performance due to more communications
  - Or suffering from memory transport issue in NUMA? Need to check
- Question: Does each 30 MB(or 3 GB) get assigned in memory closest to core that processes it?
  - Solution : numactl is used with mpirun, but the result does not improve much!
  - Question: Is numactl actually doing something?



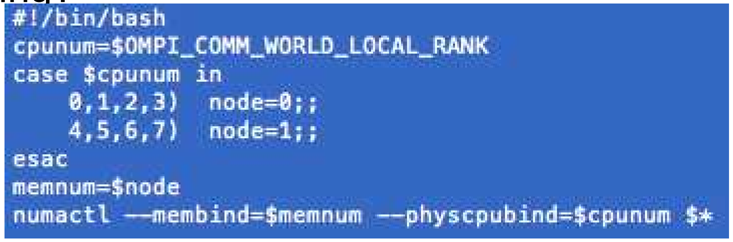
```
== Table 1 Before numactl====
           200MB   2GB   12GB
-----------------------------
2 cores    4.150
-----------------------------
4 cores    8.261
-----------------------------
8 cores    14.74  8.062  7.073
=============================

== Table 2 Using numactl=====
           200MB   2GB    12GB
-----------------------------
2 cores    4.157
-----------------------------
4 cores    8.267
-----------------------------
8 cores    14.72  8.248  7.153
```



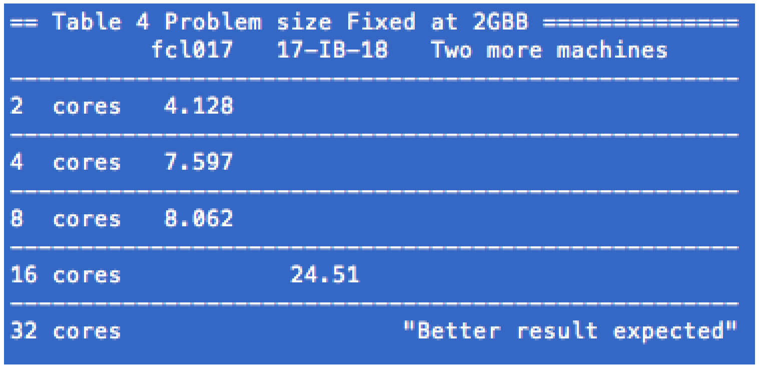
```
== numactl mapping =====
numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3
node 0 size: 12278 MB

node 1 cpus: 4 5 6 7
node 1 size: 12288 MB
```



```
== Table 3 ==============
             200MB    2 GB
-------------------------
(opposite)   14.47    5.336
8 cores      14.78    8.073
(numactl)    14.94    8.229
=========================
```



```
#!/bin/bash
cpunum=$OMPI_COMM_WORLD_LOCAL_RANK
case $cpunum in
    0,1,2,3)  node=0;;
    4,5,6,7)  node=1;;
esac
memnum=$node
numactl --membind=$memnum --physcpubind=$cpunum $*
```

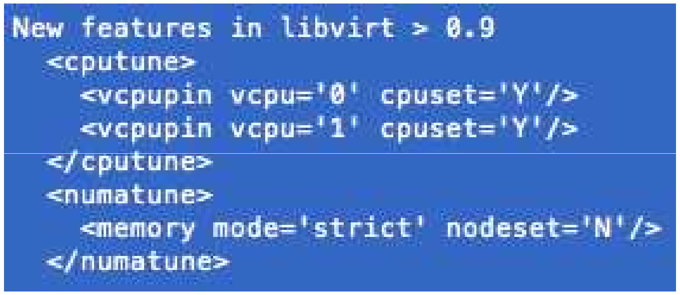


```
== Table 4 Problem size Fixed at 2GBB ==============
           fcl017   17-IB-18   Two more machines
----------------------------------------------------
2  cores   4.128
----------------------------------------------------
4  cores   7.597
----------------------------------------------------
8  cores   8.062
----------------------------------------------------
16 cores             24.51
----------------------------------------------------
32 cores                       "Better result expected"
```

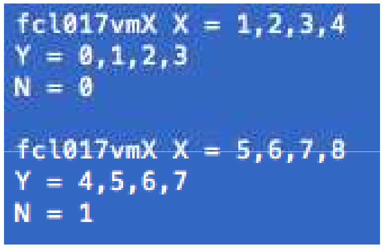
Task 3 : Test with HPL on MPI

HPL Test 2: In Virtual Machines

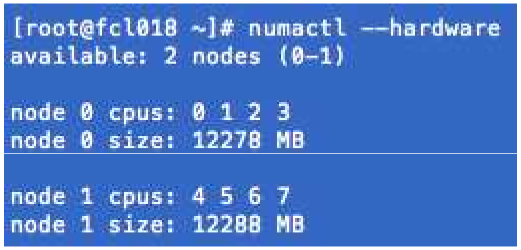
- Next, same HPL test in virtual machines
- How to pin each of 16 VMs on one specific core?
- Again numactl is used but now wrapped by libvirt
  - New elements : <cpu tune>, <numa tune> in LV 0.9.X



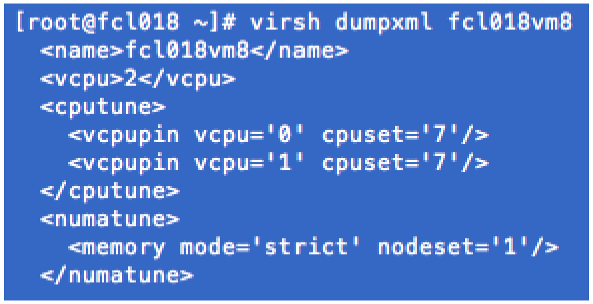
```
New features in libvirt > 0.9
  <cputune>
    <vcpupin vcpu='0' cpuset='Y'/>
    <vcpupin vcpu='1' cpuset='Y'/>
  </cputune>
  <numatune>
    <memory mode='strict' nodeset='N'/>
  </numatune>
```



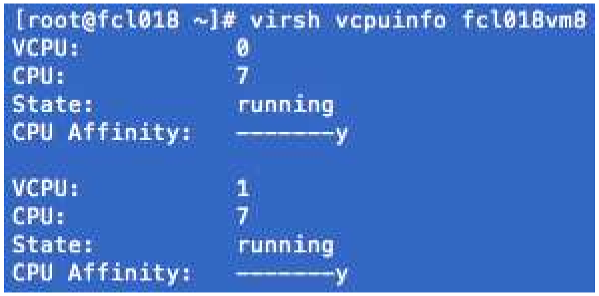
```
fcl017vmX X = 1,2,3,4
Y = 0,1,2,3
N = 0

fcl017vmX X = 5,6,7,8
Y = 4,5,6,7
N = 1
```



```
[root@fcl018 ~]# numactl --hardware
available: 2 nodes (0-1)

node 0 cpus: 0 1 2 3
node 0 size: 12278 MB

node 1 cpus: 4 5 6 7
node 1 size: 12288 MB
```



```
[root@fcl018 ~]# virsh dumpxml fcl018vm8
  <name>fcl018vm8</name>
  <vcpu>2</vcpu>
  <cputune>
    <vcpupin vcpu='0' cpuset='7'/>
    <vcpupin vcpu='1' cpuset='7'/>
  </cputune>
  <numatune>
    <memory mode='strict' nodeset='1'/>
  </numatune>
```



```
[root@fcl018 ~]# virsh vcpuinfo fcl018vm8
VCPU:           0
CPU:            7
State:          running
CPU Affinity:   -------y

VCPU:           1
CPU:            7
State:          running
CPU Affinity:   -------y
```

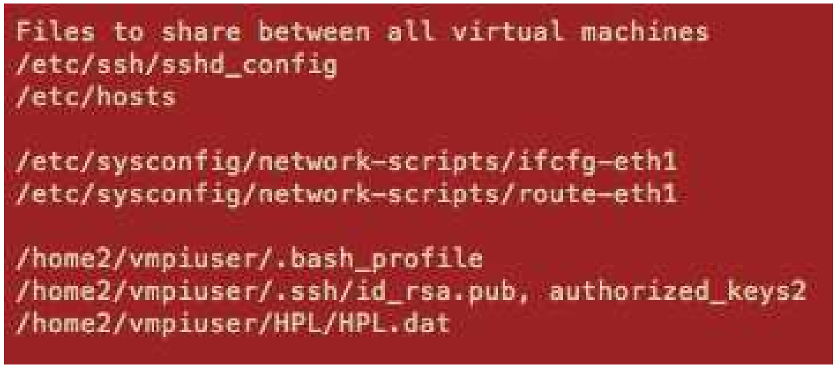
Currently I am tuning this configuration to extract meaningful results from running HPL on 16 virtual machines compared to 16 processes

Plans: Now It's Optimization

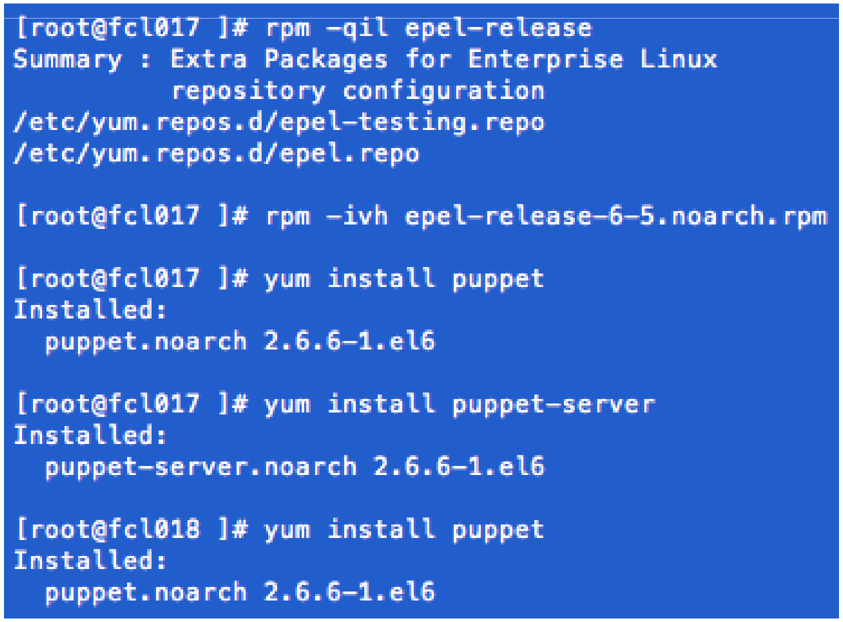
- Performance Optimization
  - – NUMA well controlled by numactl?
  - – SRIOV can do better than KVM network
  - – Inter-VM, is MPI the best to use SHM?
    - How about Multithreading via OpenMP?
- Management Optimization
  - – Virtual Machines : OpenNebula
    - Front-end on fcl017 with 16 cluster nodes(VMs)
  - – System/User File Sharing
    - Puppet : Configuration Manager

Plans: Optimizations

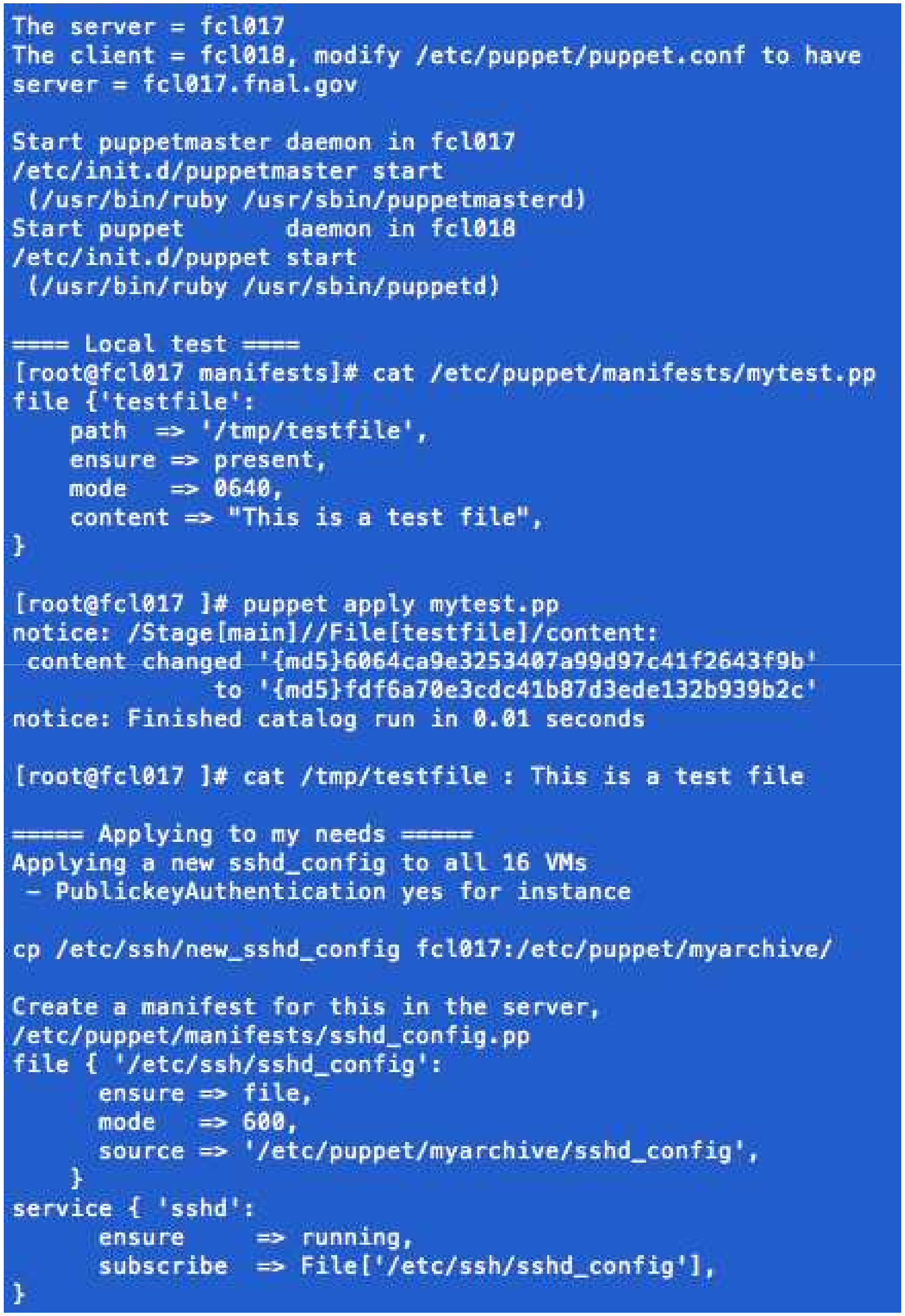
Puppet Test for File Sharing



```
Files to share between all virtual machines
/etc/ssh/sshd_config
/etc/hosts

/etc/sysconfig/network-scripts/ifcfg-eth1
/etc/sysconfig/network-scripts/route-eth1

/home2/vmpiuser/.bash_profile
/home2/vmpiuser/.ssh/id_rsa.pub, authorized_keys2
/home2/vmpiuser/HPL/HPL.dat
```



```
[root@fcl017 ]# rpm -qil epel-release
Summary : Extra Packages for Enterprise Linux
          repository configuration
/etc/yum.repos.d/epel-testing.repo
/etc/yum.repos.d/epel.repo

[root@fcl017 ]# rpm -ivh epel-release-6-5.noarch.rpm

[root@fcl017 ]# yum install puppet
Installed:
  puppet.noarch 2.6.6-1.el6

[root@fcl017 ]# yum install puppet-server
Installed:
  puppet-server.noarch 2.6.6-1.el6

[root@fcl018 ]# yum install puppet
Installed:
  puppet.noarch 2.6.6-1.el6
```

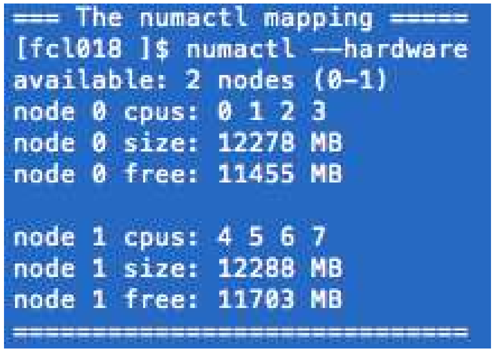


```
The server = fcl017
The client = fcl018, modify /etc/puppet/puppet.conf to have
server = fcl017.fnal.gov

Start puppetmaster daemon in fcl017
/etc/init.d/puppetmaster start
 (/usr/bin/ruby /usr/sbin/puppetmasterd)
Start puppet       daemon in fcl018
/etc/init.d/puppet start
 (/usr/bin/ruby /usr/sbin/puppetd)

==== Local test ====
[root@fcl017 manifests]# cat /etc/puppet/manifests/mytest.pp
file {'testfile':
    path    => '/tmp/testfile',
    ensure => present,
    mode    => 0640,
    content => "This is a test file",
}

[root@fcl017 ]# puppet apply mytest.pp
notice: /Stage[main]//File[testfile]/content:
 content changed '{md5}6064ca9e3253407a99d97c41f2643f9b'
             to '{md5}fdf6a70e3cdc41b87d3ede132b939b2c'
notice: Finished catalog run in 0.01 seconds

[root@fcl017 ]# cat /tmp/testfile : This is a test file

===== Applying to my needs =====
Applying a new sshd_config to all 16 VMs
 - PublickeyAuthentication yes for instance

cp /etc/ssh/new_sshd_config fcl017:/etc/puppet/myarchive/

Create a manifest for this in the server,
/etc/puppet/manifests/sshd_config.pp
file { '/etc/ssh/sshd_config':
      ensure => file,
      mode    => 600,
      source => '/etc/puppet/myarchive/sshd_config',
    }
service { 'sshd':
      ensure     => running,
      subscribe  => File['/etc/ssh/sshd_config'],
}
```

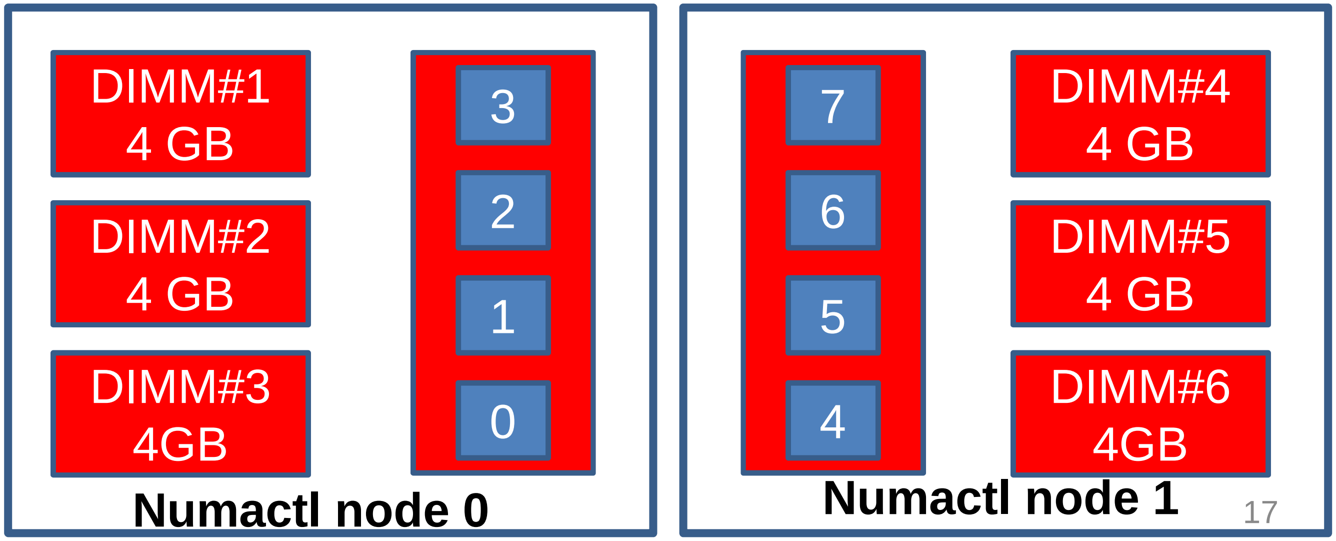
Plans: Optimizations

Optimization 1 : NUMA

- Question: Does numactl do the best?
- Question: How is the mapping decided?
  - The numactl source code : just looks at /sys/devices/system/node
  - Is this mapping the most optimized?
  - Is it equivalent to the real configuration?
  - Question: Any way to verify the effect of numactl?
    - Don Holmgren old experience writing a program to check the mapping of virtual memory space of a process to physical memory.

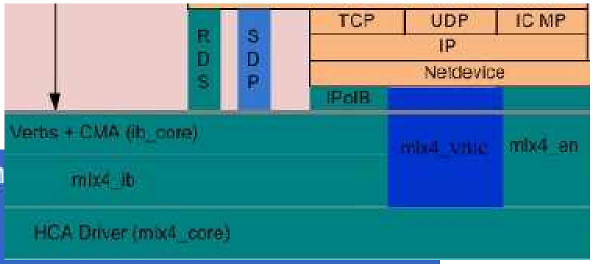


```
=== The numactl mapping ======
[fcl018 ]$ numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3
node 0 size: 12278 MB
node 0 free: 11455 MB

node 1 cpus: 4 5 6 7
node 1 size: 12288 MB
node 1 free: 11703 MB
==============================
```



Optimization 2 : SRIOV

- Two things I can do before Mellanox SRIOV
  1. Use Intel GbE 82576 controllers
     - modprobe igb max_vfs =2



```
08:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection
08:10.0 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
08:10.2 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)

02:00.0 InfiniBand: Mellanox Technologies MT26418 [ConnectX VPI PCIe 2.0 5GT/s - IB DDR / 10GigE] (rev b0)
```

```
[root@fcl017 ~]# virsh edit fcl017vm1
 <hostdev mode='subsystem' type='pci' managed='yes'>
 <address domain='0x0000' bus='0x08' slot='0x10' function='0x0'/>
[root@fcl017 ~]# ip link set eth0 vf 0 mac 54:52:00:02:17:12
```

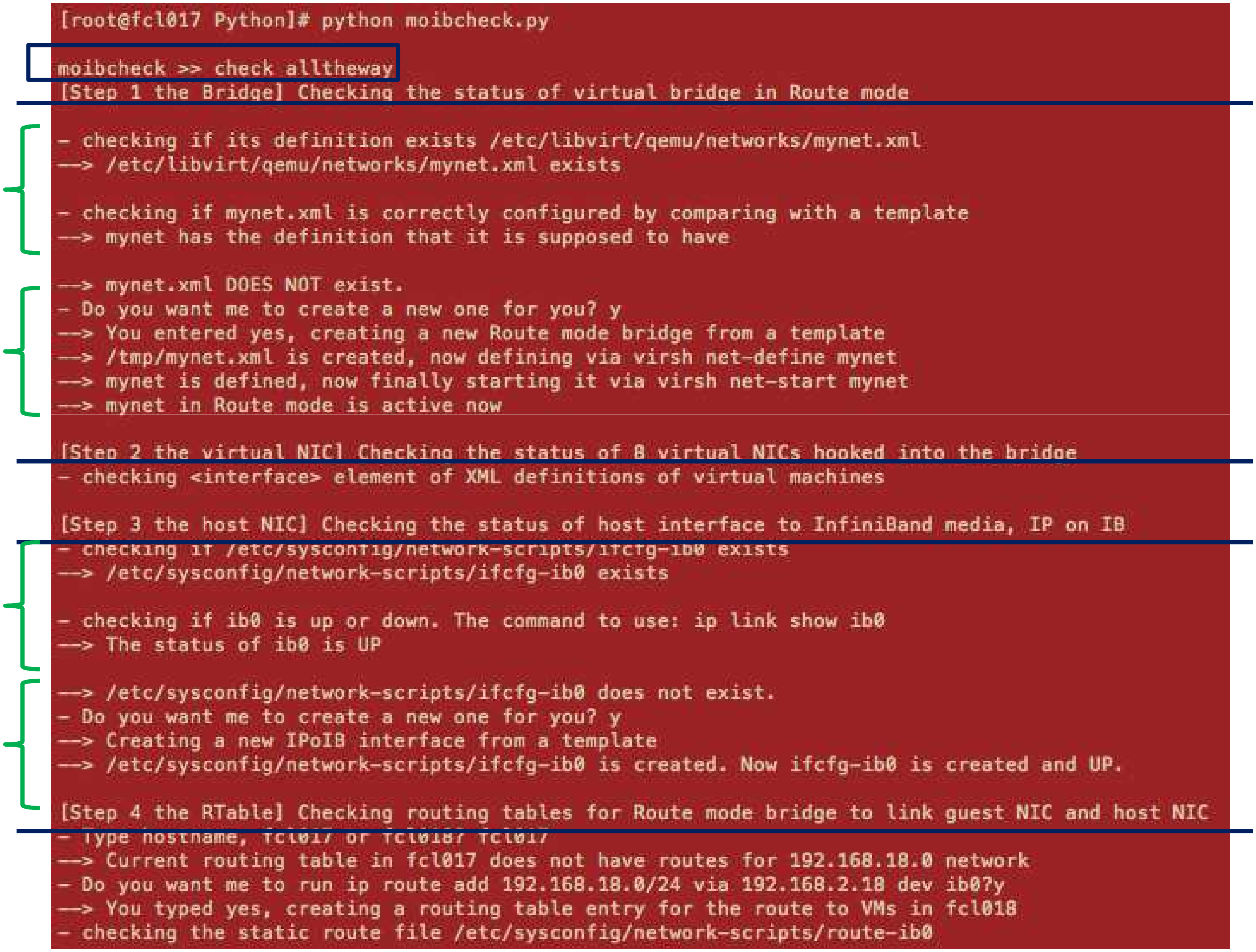
```
fcl017vm1 /etc/udev/rules.d/70-persistent-net.rules
ATTR{address}=="54:52:00:02:17:12", NAME="eth0"
```

  2. Trying SRIOV with our current Mellanox adapters
     - Says mlx4_core does it
     - modprobe mlx4_core max_vfs=2 : crashes
     - /rpmbuild/SOURCES/ofa_kernel/drivers/net/mlx4/main.c
       - to find correct name if any, unsuccessful
     - A patch in RHEL5.5 to enable SRIOV in mlx4_core

Optimization 3 : MultiThreading

- Will write a small parallel program that can be implemented both by MPI and OpenMP
- And compare

Overall Control by a python script



```
[root@fcl017 Python]# python moibcheck.py

moibcheck >> check alltheway
[Step 1 the Bridge] Checking the status of virtual bridge in Route mode

- checking if its definition exists /etc/libvirt/qemu/networks/mynet.xml
--> /etc/libvirt/qemu/networks/mynet.xml exists

- checking if mynet.xml is correctly configured by comparing with a template
--> mynet has the definition that it is supposed to have

--> mynet.xml DOES NOT exist.
- Do you want me to create a new one for you? y
--> You entered yes, creating a new Route mode bridge from a template
--> /tmp/mynet.xml is created, now defining via virsh net-define mynet
--> mynet is defined, now finally starting it via virsh net-start mynet
--> mynet in Route mode is active now

[Step 2 the virtual NIC] Checking the status of 8 virtual NICs hooked into the bridge
- checking <interface> element of XML definitions of virtual machines

[Step 3 the host NIC] Checking the status of host interface to InfiniBand media, IP on IB
- checking if /etc/sysconfig/network-scripts/ifcfg-ib0 exists
--> /etc/sysconfig/network-scripts/ifcfg-ib0 exists

- checking if ib0 is up or down. The command to use: ip link show ib0
--> The status of ib0 is UP

--> /etc/sysconfig/network-scripts/ifcfg-ib0 does not exist.
- Do you want me to create a new one for you? y
--> Creating a new IPoIB interface from a template
--> /etc/sysconfig/network-scripts/ifcfg-ib0 is created. Now ifcfg-ib0 is created and UP.

[Step 4 the RTable] Checking routing tables for Route mode bridge to link guest NIC and host NIC
- Type hostname, fcl017 or fcl018? fcl017
--> Current routing table in fcl017 does not have routes for 192.168.18.0 network
- Do you want me to run ip route add 192.168.18.0/24 via 192.168.2.18 dev ib0?y
--> You typed yes, creating a routing table entry for the route to VMs in fcl018
- checking the static route file /etc/sysconfig/network-scripts/route-ib0
```

Summary

- Cluster of virtual machines for MPI is ready!
  - For now with KVM(virtual networks) on IB
  - Later with SRIOV on InfiniBand
- Now tuning the cluster using HPLinpack
- Looking into OpenNebula and Puppet
  - For management optimization
- Python script for initialization/maintenance
- Technical Note is being prepared now